BioE/MCB/PMB C146/C246: Topics in Computational Biology and Genomics

Lecture 2) Sequence Evolution

23 Jan 2003

Scribe: Aaron Garnett *Reader*: Claire Collins

Administrative notes

There is a Perl class being taught in the computer science department. The class web site could be useful and the address is http://www.ocf.berkeley.edu/~darin/perl.

Homework 1 should be e-mailed to derek@rana.lbl.gov by Monday January 27 by 5PM and a printout should be turned in on Tuesday in class.

Clarification of homework 1 problem 5: you should compare the DNA sequences after putting them through the 1000 generations and count how many times a change is made at each position over the 10 trials.

Part of your grade is based on class participation which includes doing the scribe notes when it's your turn, participating in class discussions and attending class discussions.

Interesting fact

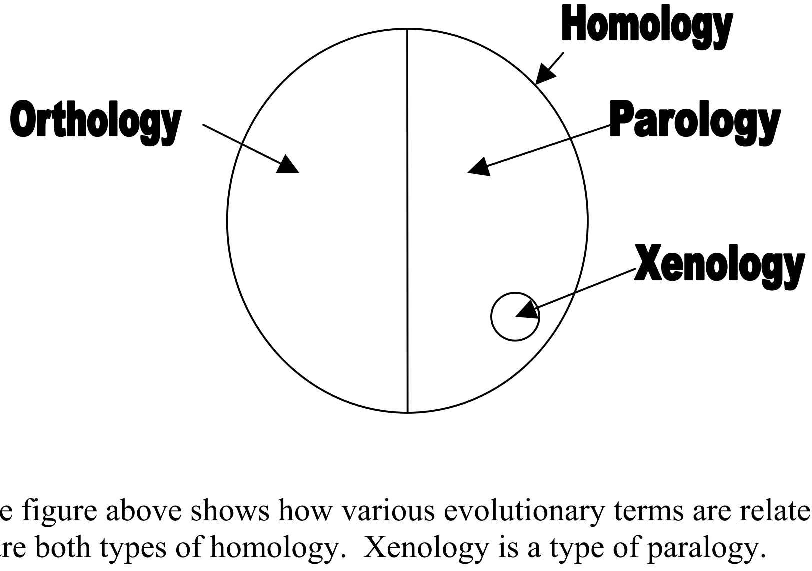
400,000,000,000,000 BLAST comparisons are made each year at NCBI.

Rates of evolution

-We can infer homology for sequences which diverged up to 15 billion years ago

-For quickly evolving proteins we can't see as far back. For example, the Bruno peptides are under quickly changing selective pressure.

Evolutionary relationships



The figure above shows how various evolutionary terms are related. Orthology and paralogy are both types of homology. Xenology is a type of paralogy.

Homoplasy

Definition: The relationship between any two identical character states that must have arisen independently, given a specific phylogenetic tree.

Examples:

1. The wings of birds and bats evolved independently, but they both evolved them from forelimbs.

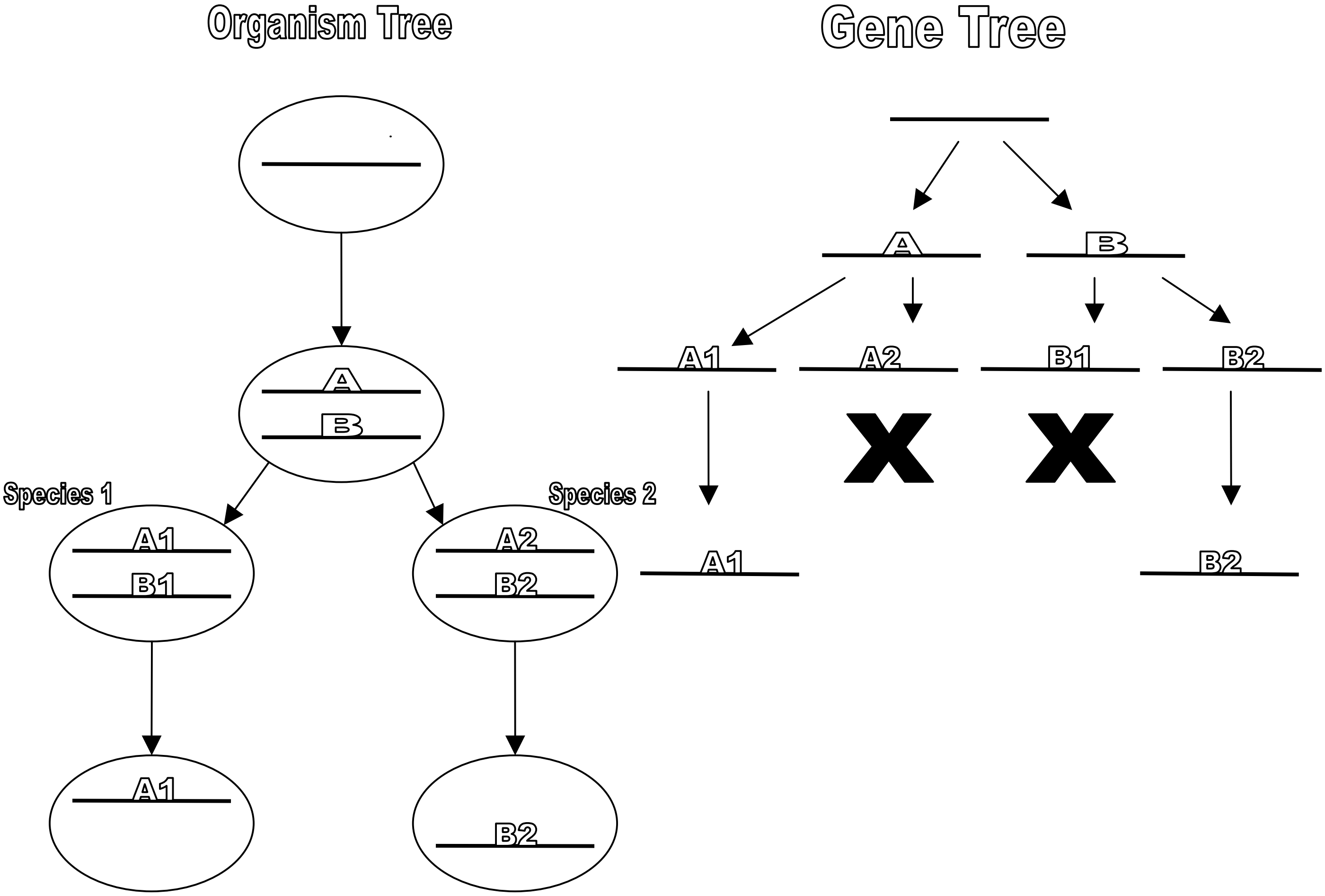
2. The lysozymes in Langer monkeys and cows evolved independently, but they are both involved in breaking down gut bacteria. Both have evolved to survive in an acid rich environment. This shows some level of convergent evolution.

Challenging cases of gene relationship identification

Alleles vs. Genes

Should alleles be considered paralogs or orthologs? Fitch's paper makes the case for paralogs.

Two independent gene losses



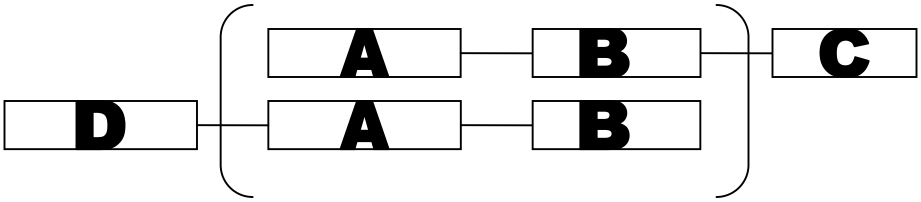
See the above figure. If a gene is duplicated in a species (giving genes A and B) and the species undergoes speciation (giving species 1 and 2 in the figure) and each species looses a different gene (species 1 loses gene A and species 2 looses gene B), the two remaining genes (genes A1 and B2) are paralogs, but could appear to be orthologs if subject to sequence comparison.

There is a real state for the relationship between two genes (homology, analogy, paralogy, etc), we just have a hard time figuring it out sometimes.

To determine if genes are paralogous or orthologous, look up the tree and check if the genes separated at a duplication or a speciation event. If they diverged at a speciation event, they are orthologs. If they diverged at a duplication event, they are paralogs.

Partial Homology

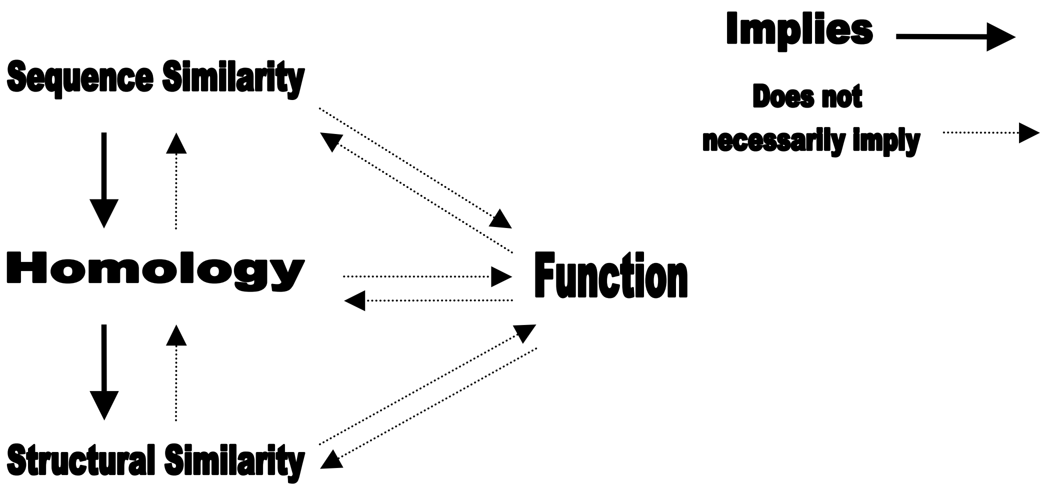
Sometimes seen in muti-domain proteins. Two proteins may contain related domains, while other domains may be unrelated. In the figure below both proteins contain domains A and B, but differ in their third domains.



The argument over ancestor determination

Neurath, Winter and Walsh (1968) argued there is no way to tell if two sequences are homologous or orthologous.

Fitch (1970) argued you should be able to predict how many mutations it took to get from the ancestor to the present day sequences. If you assume the minimal number of mutations took place and build a tree based on differences in the sequences you can determine whether the sequences are diverging or converging. This method is based purely on an evolutionary tree, not sequence similarity. However, experience with building a bunch of these trees has shown us that sequence similarity implies homology. This is not true when looking at protein structures, non-homologous proteins can have similar structures. Homology does not imply sequence similarity. Homology normally implies structural similarity. The relationship of these properties with function are uncertain. See the figure below.



Examples of the relationships among these properties

TIM barrels

Bacterial luciferase and non-fluorescent flavoprotein both contain TIM barrels, have high degrees of sequence and structural similarity, and are clearly homologous, but they have different functions.

FNIII and Ig domains

These proteins are structurally similar, but don't appear to be homologous. They have little sequence similarity and different functions.

Proteases

Proteases can be divided into several groups. They have a wide range of structures, but similar functions.

Sequence Alignment

Sequence alignment is important because it allows us to know which bases correspond to each other in different sequences. Consider these two sequences:

```
ACGTCG

AGCGG
```

When aligned they give allow us to predict an ancestral sequence:

```
ACGTCG

A-GCGG

ACGCCG Ancestral sequence
```

Dot Plots

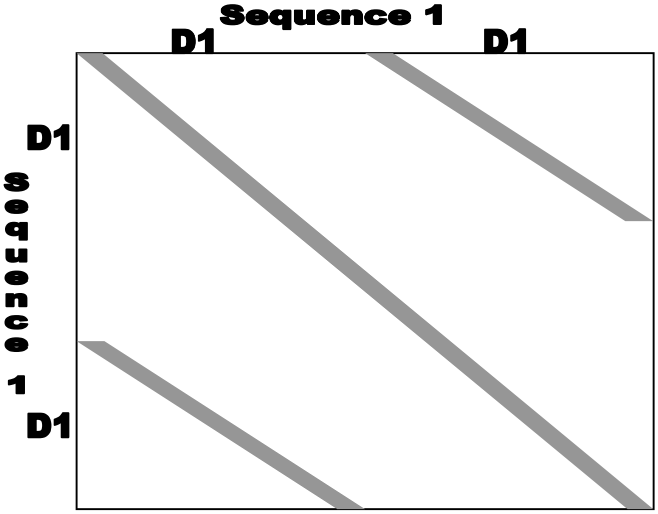
This is one way to find an alignment and is important because it is old (has historical value), simple, and is useful for looking at real sequences and understanding modern alignment methods.

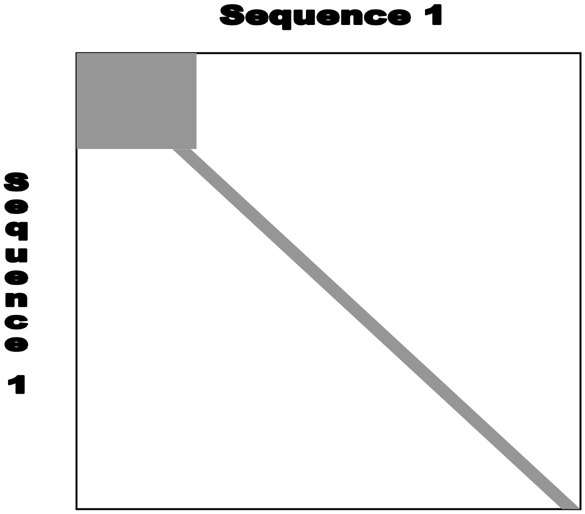
To do a dot plot, make a matrix with one sequence written across the top and the other down the side. This gives you one square for each pair of bases that can be aligned. If two bases are the same, put a mark in the square for those two.

	A	C	G	T	C	G
A	x					
G			x			x
C		x			x	
G			x			x
G			x			x

Now look for diagonals in the plot, these correspond to regions of ungapped alignment.

It can also be useful to do a dot plot using the same sequence on both axes. If you see several diagonals in the plot, these could correspond to repeats. In the example below on the left, the subsequence D1 is repeated in the sequence. If you see a rectangular area with a lot of scattered dots, as in the example on the right, this indicates a region of low sequence complexity. In other words, the region has a high level of repetition and uses a low number of letters.





There are different types of dot plots with different levels of sophistication. Dot plot rules can be used to find sequence similarity as well as identity. For example, you could draw a dot in a position corresponding to two similar amino acids. You can also use a "sliding window" technique in which you compare overlapping sections, or windows, of the sequences based on each position in the sequence and give each window a similarity score.

Dot plots are a nice tool because they give a visual overview of sequence similarity.

An introduction to sequence alignment using dynamic programming

Try to align these two sequences:

```
ATCGCGAACG

ACGCGTAACT
```

It is pretty easy to do visually, but there are a whole bunch of possible alignments which we need to sort through to find the correct one if we want to do it systematically.

To compare the quality of alignments, we need a scoring system. We'll start with this one:

Match=+1

Mismatch=-1

Gap=-2

In this case here is the optimal alignment looks like this

```
ATCGCG-AACG
| |||| |||X
A-CGCGTAACT
```

This alignment has 8 matches 1 mismatch and 2 gaps, giving a score of +3. If we change the gap penalty from -2 to -5, this alignment has a score of -3, and will no longer be the optimal alignment. This example illustrates that identity of the optimal alignment is sensitive to the scoring system.

Now that we have a scoring system we need an efficient way to sort through the possible alignments and find the best alignment. To do this we consider what would happen if we forced each base in sequence 1 to align with each base in sequence 2. Here we need some notation:

i is the position of a base in the first sequence.

j is the position of a base in the second sequence.

s_{ij} is the score for aligning base i in the first sequence with base j in the second. So according to our scoring scheme s_{ij}=1 if bases i and j are the same, and s_{ij}=-1 if they don't match.

S_{ij} is the score for the optimal alignment of the sequences up to position i in the first sequence and position j in the second. For example if we are aligning the sequences above, $S_{3,4}$ is the score of the optimal alignment for the sequences up to position 3 in the first sequence (ATC) and position 4 in the second (ACGC).

g is the gap penalty.

Now we can find the score of the optimal alignment up to any point in the sequences. To do this we calculate S_{ij} for every possible combination of i and j. In other words, we force each base in sequence 1 to align with each base in sequence 2 and find the optimal alignment for the sequences up to that point. For every i and j pair you need to decide whether or not to add a gap to extend the alignment. You want make the choice that will give you the highest score for the alignment up to that point. With our notation this choice looks like this:

$$S_{ij}=\max\begin{cases} S_{i-1,j-1}+s_{ij} \text{ (i.e. extend the alignment without adding a gap)} \\ S_{i-1,j}-g \text{ (extend the alignment by adding a gap to the second sequence)} \\ S_{i,j-1}-g \text{ (add a gap to the first sequence)} \end{cases}$$

There is also the boundary condition:

$S_{0,0}=0$

In this way, for each i/j pair, the optimal alignment for the entire sequence can be seen as combination of the optimal alignment for the sequences before i and j and the optimal alignments of the sequences after i and j.

We now have a tool to break the larger sequence alignment problem into smaller sequence alignment problems. If we keep track of the optimal alignment for every i/j pair, we can determine the optimal alignment for the entire sequence and search the exponential alignment space using only n^2 calculations. This is a type to memo-ization called dynamic programming.